SQL  
  
to update: UPDATE column name, set filed = ‘value’, where id = 4;

To delete rows: DELETE from column name, where column is NULL;

To create table with constraints: CREATE TABLE name (id INTEGER PRIMARY KEY);

<https://www.codecademy.com/learn/learn-sql/modules/learn-sql-aggregate-functions/cheatsheet>

wild CARD ‘%’ BEFORE OR AFTER OR \_

WHERE NAME LIKE ‘THE %’

IF else:

SELECT name,  
CASE  
  WHEN imdb\_rating > 8 THEN 'Fantastic'  
  WHEN imdb\_rating > 6 THEN 'Poorly Received'  
  ELSE 'Avoid at All Costs'  
END AS 'Review'  
FROM movies;

Average Price Rounded by 2 decimal points

SELECT ROUND(AVG(price), 2)  
FROM fake\_apps;

select category, sum(downloads)

from fake\_apps

group by category;

select price, round(avg(downloads)), count(\*)

from fake\_apps

group by price

having count(name) > 10;

WITH previous\_query AS (

SELECT customer\_id,

   COUNT(subscription\_id) AS 'subscriptions'

FROM orders

GROUP BY customer\_id)

select customers.customer\_name,

 previous\_query.subscriptions

 from previous\_query

JOIN customers

ON customers.customer\_id = previous\_query.customer\_id;

SELECT DISTINCT column, AGG\_FUNC(*column\_or\_expression*), …

FROM mytable

JOIN another\_table

ON mytable.column = another\_table.column

WHERE *constraint\_expression*

GROUP BY column

HAVING *constraint\_expression*

ORDER BY *column* ASC/DESC

LIMIT *count* OFFSET *COUNT*;

|  |  |  |
| --- | --- | --- |
| **Operator** | **Condition** | **SQL Example** |
| =, !=, < <=, >, >= | Standard numerical operators | col\_name **!=** 4 |
| BETWEEN … AND … | Number is within range of two values (inclusive) | col\_name **BETWEEN** 1.5 **AND** 10.5 |
| NOT BETWEEN … AND … | Number is not within range of two values (inclusive) | col\_name **NOT BETWEEN** 1 **AND** 10 |
| IN (…) | Number exists in a list | col\_name **IN** (2, 4, 6) |
| NOT IN (…) | Number does not exist in a list | col\_name **NOT IN** (1, 3, 5) |

|  |  |  |
| --- | --- | --- |
| **Operator** | **Condition** | **Example** |
| = | Case sensitive exact string comparison (**notice the single equals**) | col\_name **=** "abc" |
| != or <> | Case sensitive exact string inequality comparison | col\_name **!=** "abcd" |
| LIKE | Case insensitive exact string comparison | col\_name **LIKE** "ABC" |
| NOT LIKE | Case insensitive exact string inequality comparison | col\_name **NOT LIKE** "ABCD" |
| % | Used anywhere in a string to match a sequence of zero or more characters (only with LIKE or NOT LIKE) | col\_name **LIKE** "%AT%" (matches "AT", "ATTIC", "CAT" or even "BATS") |
| \_ | Used anywhere in a string to match a single character (only with LIKE or NOT LIKE) | col\_name **LIKE** "AN\_" (matches "AND", but not "AN") |
| IN (…) | String exists in a list | col\_name **IN** ("A", "B", "C") |
| NOT IN (…) | String does not exist in a list | col\_name **NOT IN** ("D", "E", "F") |

select first\_name from patients

group by first\_name

having count(\*) = 1

select first\_name from patients

order by len(first\_name),

first\_name asc

SELECT

(SELECT count(\*) FROM patients WHERE gender='M') AS male\_count,

(SELECT count(\*) FROM patients WHERE gender='F') AS female\_count;

select patient\_id, primary\_diagnosis from admissions

group by patient\_id, primary\_diagnosis

having count(\*)>1

select first\_name, last\_name, 'Patient' as 'role' from patients

union

select first\_name, last\_name, 'Physician' as 'role' from physicians

select concat(UPPER(last\_name), ",", (LOWER(first\_name))) as 'full name' from patients

order by first\_name desc

select province\_id, sum(height) as 'sumofHeight' from patients

group by province\_id

having sum(height)>=7000

select (max(weight)-min(weight)) as "weight" from patients

where last\_name LIKE 'Maroni'

select patient\_id, nursing\_unit\_id, room, bed from admissions

where patient\_id = '542'

group by patient\_id

having MAX(admission\_date)

odd patient id, physician in or contains 2 in physician id and length of 3 for patient id

select patient\_id, attending\_physician\_id, primary\_diagnosis from admissions

where (attending\_physician\_id in (1, 5, 19) and patient\_id %2 !=0)

or (attending\_physician\_id like '%2%' and len(patient\_id)=3)

select patient\_id, weight, height,

(case

WHEN weight/(power(height/100.0, 2)) >=30 THEN 1

ELSE 0

END) AS isObese

FROM patients;

select

case

when patient\_id %2=0 then 'yes'

else 'no'

end as has\_insurance,

sum(case when patient\_id%2=0 then '10'

else '50'

end) as cost\_after\_insurance

from admissions

group by has\_insurance

select province\_name from patients

join provinces

on provinces.province\_id = patients.province\_id

group by province\_name

having sum(gender = 'M') > Sum(gender = 'F')

select \* from patients

where first\_name like '\_\_r%'

and month(birth\_date) in (2,5,12)

and gender like 'F'

and weight between 60 and 80

and patient\_id % 2 != 0

and city like 'Halifax'

//goes to previous count(LAG)

select admission\_date, count(\*) as admission\_day, count(\*)-lag(count(\*), 1) over() as admission\_count\_change from admissions

group by admission\_date